

AD-A161 827

AD



US ARMY
MATERIEL
COMMAND

MEMORANDUM REPORT BRL-MR-3473

AN EXPERT SYSTEM FOR PREDICTING COMPONENT KILL PROBABILITIES

Richard A. Helfman
John C. Saccenti
Richard E. Kinsler
J. Robert Suckling

October 1985

DTIC
FLECTE
DEC 0 2 1985
S D

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

US ARMY BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

FILE COPY

85 11 25 124

Destroy this report when it is no longer needed.
Do not return it to the originator.

Additional copies of this report may be obtained
from the National Technical Information Service,
U. S. Department of Commerce, Springfield, Virginia
22161.

The findings in this report are not to be construed as an official
Department of the Army position, unless so designated by other
authorized documents.

The use of trade names or manufacturers' names in this report
does not constitute indorsement of any commercial product.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER MEMORANDUM REPORT BRL-MR-3473	2. GOVT ACCESSION NO. AD-A161827	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AN EXPERT SYSTEM FOR PREDICTING COMPONENT KILL PROBABILITIES		5. TYPE OF REPORT & PERIOD COVERED FINAL
7. AUTHOR(s) Richard A. Helfman John C. Saccenti Richard E. Kinsler J. Robert Suckling		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS US Army Ballistic Research Laboratory ATTN: SLCBR-VL Aberdeen Proving Ground, MD 21005-5066		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 1L162618AH80
11. CONTROLLING OFFICE NAME AND ADDRESS US Army Ballistic Research Laboratory ATTN: SLCBR-DD-T Aberdeen Proving Ground, MD 21005-5066		12. REPORT DATE October 1985
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES 38
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution is unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Expert System Components Vulnerability		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) An expert system (ES) is one which contains a large amount of high level knowledge about a specific domain and it can apply that knowledge to achieve acceptable levels of performance in problem solving. Target application areas for ES include those areas for which there are perhaps a few simple underlying theories and a rather large accumulation of high level knowledge considered of paramount importance in the analytic problem solving process. (continued)		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

The analytical methodology developed at the BRL for predicting the probability of rendering an Army system component non-functional (killing the component) relies heavily on analyst experience and thus contains a degree of subjectivity. Expert guidance in component defeat analysis is needed to provide consistency and lessen the degree of subjectivity. With the assistance of an expert, an ES has been developed to codify and preserve the "lines of reasoning" associated with determining the probability of kill (P k/h) given that a system component is hit by a selected battlefield threat. The goal of the project was to build an ES with the characteristics of:

- consistent and correct predictive capabilities,
- making the non-expert user as productive as the expert,
- preservation of corporate knowledge.

The probability data thus produced utilizing the ES is employed with subsequent methodology to determine system vulnerability, to reduce system vulnerability and to enhance system survivability.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

CONTENTS

I.	Introduction.....	1
II.	Objectives.....	1
III.	Expert System.....	2
IV.	P (k/h) Methodology Background.....	3
V.	BRL Expert System.....	6
1.	Obtain and validate the physical properties of the component.....	7
2.	Derive the input parameters to COMPKIL.....	8
a.	Presented area.....	8
b.	Sensitive area.....	8
c.	Hole diameter.....	9
d.	Barriers, materials, thicknesses.....	9
3.	Run COMPKIL and prepare inputs for VAST.....	9
VI.	System Details.....	10
VII.	Code Details.....	10
1.	Working Memory.....	10
2.	Knowledge Base.....	11
3.	Sample Rules.....	11
VIII.	Corporate Memory.....	13
IX.	Current Status.....	13
X.	Future Directions.....	13
XI.	Conclusion.....	14
APPENDIX A:	Sample ES Session.....	15
APPENDIX B:	Sample Rules.....	23
APPENDIX C:	Help Feature.....	33
DISTRIBUTION LIST	37

I. INTRODUCTION

The vulnerability of a target is a measure of the susceptibility of that target to damage by a specified damage producing mechanism. The damage level is related to a given preselected kill criterion that defines the rapidity and/or the degree to which the target will suffer loss of function. For a given kill criterion, the target must be analyzed to determine:

1. Those components (termed critical) essential to the proper functioning of the target.
2. The level of damage to each critical component required to satisfy the kill criterion.

For example, consider a tank. A reasonable kill criteria might be loss of mobility (M-kill) within a specified time (say 20 minutes). It can be seen that loss of the fuel system will result in an M-kill, since the tank will stop. However, loss of the fire extinguisher system would not result in M-kill, since in general, mobility would not be affected.

Some of the critical components of the fuel system are the fuel tank, fuel line and the fuel pump. The kill requirement for the fuel tank would be a hole of sufficient size to drain the tank within the time specified by the kill criterion. Similarly, a hole in the fuel line of sufficient size to prevent the fuel from being pumped to the carburetor would be the kill requirement for the fuel line.

The conditional kill probability P (k/h), for a critical component, is defined as the probability of rendering the component nonfunctional given that it is hit.

II. OBJECTIVES

The purpose of this ^{document} effort was to develop an expert system to :

- model both the numeric and non-numeric aspects of component defeat analysis;
- capture the corporate knowledge involved in appraising component defeat;
- incorporate this knowledge into a usable knowledge/information based computer model and
- demonstrate the utility of such a system.

The system should accommodate all types of users, from novice to expert. It should handle all of the routine analyses and as much of the unusual or unexpected as possible. Furthermore, it should aid the analyst/expert rather than replace him.

III. EXPERT SYSTEM

An expert system (ES) is a computer program which contains a large amount of explicitly high-level knowledge about a limited domain and is able to apply that knowledge to achieve acceptable levels of problem solving in that domain. The strength of an ES is in dealing with situations which have many different configurations and in correctly identifying which one is valid based on available information. This often happens in domains where there is no simple underlying theory but rather only a large accumulated body of high-level knowledge about the domain. While some expert systems have been in existence for 20 years, it is only recently that the technology has become sufficiently well-understood to give confidence that the expert system approach itself is ready for a variety of practical applications.

The most common ES architecture is based on the production rule formalism. This formalism consists of three parts:

1. a set of rules (IF condition THEN action),
2. a data base,
3. a rule interpreter.

The rules represent the knowledge of the expert and form the "knowledge base" of the system. Each rule is composed of an antecedent part (the "if" part) and a consequent part (the "then" part). The data base, sometimes called working memory, may contain general information about the domain and also specific information about the current problem being solved. In a production rule system, the rule interpreter attempts to find a rule whose antecedent (if) part matches the data base. Whenever this happens, the consequent (then) part of the rule is used to add to or modify the data base, and we say that the rule has "fired."

Target application areas for expert systems are those for which algorithmic solutions do not exist. The modus operandi is rooted in symbolic representation (i.e. not arithmetic), and processes where extensive computer searches through elaborate mazes are performed. Utilization of implicit solution steps (heuristic, or rule-of-thumb) is the direction taken in ES programming in contrast to algorithmic solutions continually sought to produce exact answers in conventional programming techniques. Control structure (operational flow) found in conventional programming always becomes heavily integrated with the information

and thus difficult to modify. The ES approach, in contrast, strives to separate control from the knowledge domain thus providing flexibility and ease in updating the knowledge domain, modifying and enlarging it as required.

IV. P (K/H) METHODOLOGY BACKGROUND

A target vulnerability analysis is a key element of performance evaluation in programs designed to assess the effectiveness of munitions against targets, and also effectiveness of candidate measures potentially useful in providing vulnerability reduction. It is a complex process involving many analytic tasks addressing: threats and basic damage mechanisms, a computerized target description, a functional analysis of the target and a defeat criteria characterization.

Several computerized analytic methodologies are currently applied to the problem of assessing total target vulnerability. These include models to evaluate fragment perforation of materials, probability of kill given a hit (COMPKIL), methods of computer modeling targets (GIFT) and total target vulnerable area assessments (VAST).

An analytical methodology has been developed at BRL for predicting the probability of rendering a component nonfunctional (killing the component) with the impact of a single penetrator. The total target is essentially "split apart" to look at individual components, and then the probability of defeating each such component, given that it is hit, is determined. The basic premise of the component methodology is that the probability that a random hit by a penetrator of known mass and velocity will render a component nonfunctional is determined by a ratio of the sensitive area to the presented area of the component, provided the penetrator makes a large enough hole in the component. The presented area is determined by projection of the silhouette of the component onto the six faces of an enclosing cube. See Figure 1. The parts of the presented area where impacts can potentially kill the component are defined as the sensitive areas. See Figure 2.



Accession For	
NTIS ORA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Dist. Edition	
Availability Codes	
Dist	Avail. and/or Subject
A-1	

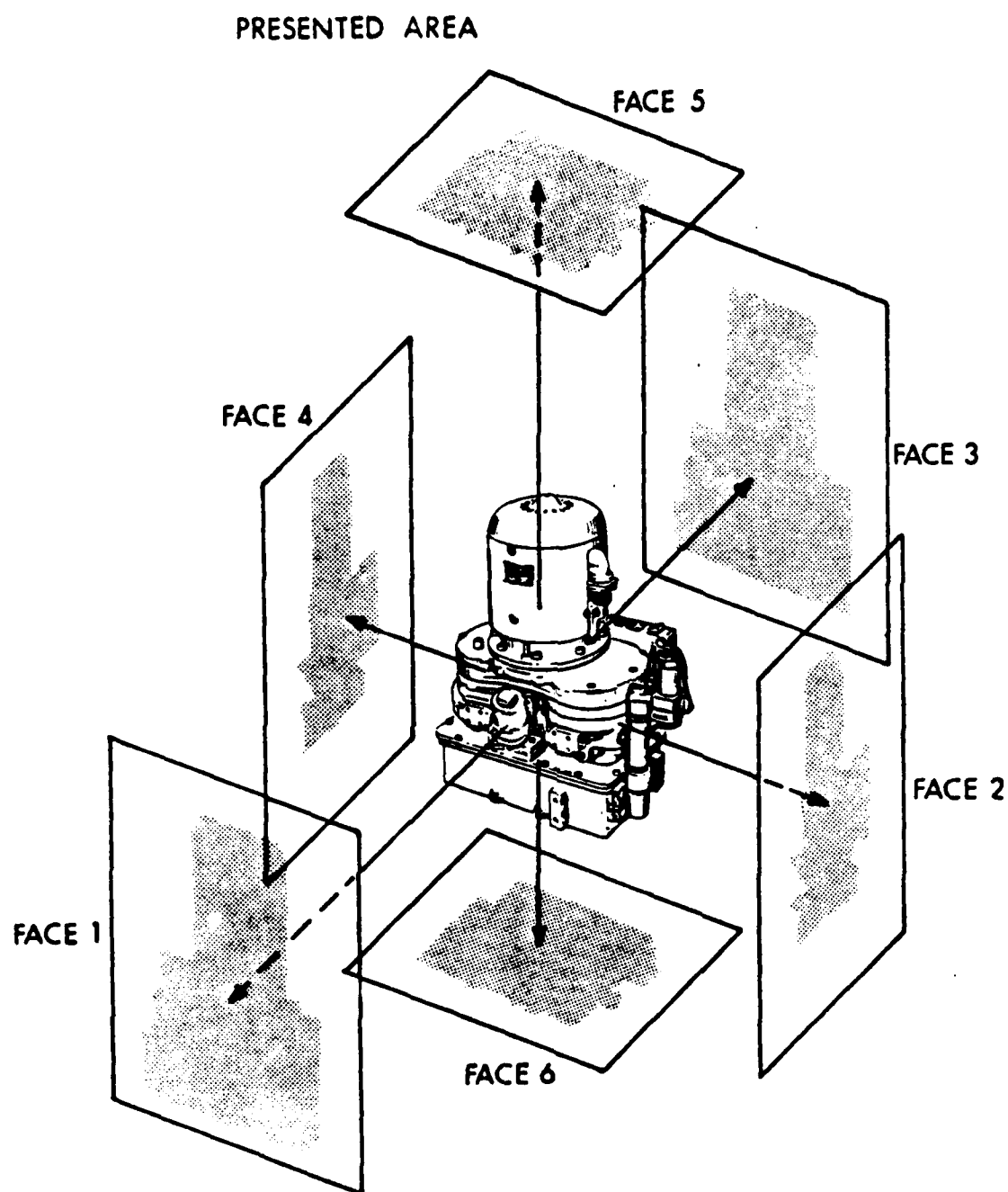


Figure 1. Presented Area

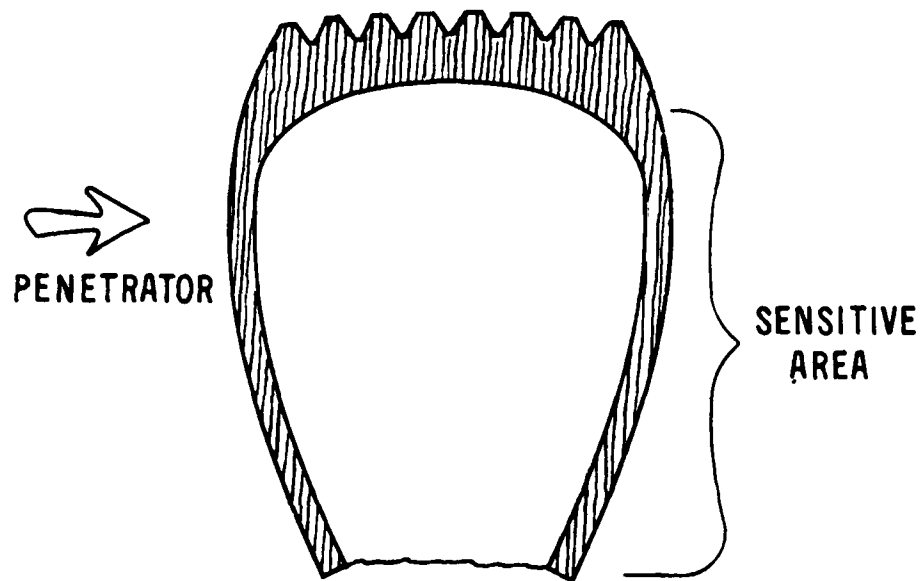


Figure 2. Sensitive Area

The program COMPKIL, written to compute the probability of conditional kill for components impacted by fragments of known mass and velocity, requires the following input parameters for each of the six component faces:

- a. presented area,
- b. sensitive area,
- c. hole diameter required to render component nonfunctional,
- d. number of barriers in the path,
- e. material type of each barrier and
- f. thickness of each barrier.

A degree of subjectivity can exist among analysts who must choose the appropriate values for the input parameters to COMPKIL, (especially parameters b and c above), and results can vary greatly depending on the experience of the analyst. Experts with extensive knowledge of the subject of component defeat generally provide a consistency needed to lessen the degree of subjectivity.

V. BRL EXPERT SYSTEM

An ES has been developed with the assistance of an expert in component defeat to calculate kill probabilities for components in armored or light vehicles. The ES currently is capable of recognizing and processing the subclass of components called control rods (cables, rods, drive shafts, hydraulics) for ground vehicles. Once the component's physical properties are known to the ES, the input parameters to COMPKIL are derived, COMPKIL is run with those parameters, and the results are displayed to the user.

A P (K/H) ES analysis is performed in three stages:

1. Obtain and validate the physical properties of the component,
2. Derive the input parameters to COMPKIL (presented area, sensitive area, hole size, etc.) and
3. Feed the results to COMPKIL and prepare inputs for VAST.

The expert's knowledge is used in stage 1 to validate the component's description (provided by the less-than-expert user) and in stage 2 to prepare the needed input parameters to COMPKIL. An elaboration of each of these stages follows.

1. Obtain and validate the physical properties of the component

The first step in the analysis is to specify the component's description. This is currently done through ES queries to the user. The user will respond by either selecting an available candidate from a menu, or accepting default settings, or by typing in data. (The ES will soon be able to read a computerized three dimensional target description, thus bypassing some of the queries to the user.)

The next step is to validate the component information. Errors can occur in at least two ways: typing and improper component description. Typing errors can occasionally happen while the user is responding to the ES queries, and the user may not always catch them. Also, a non-expert user might have a component description that is incorrect, yet not realize it.

The ES uses its knowledge base to validate the component's properties. It must be able to tell whether a component has been poorly or improperly described. For example, a control rod moved by human wrist action is usually thin, a drive shaft is usually hollow, a control cable in an armored vehicle is generally not made out of aluminum, and a hydraulic cylinder should have a length much greater than its diameter.

There are four possible outcomes from the validation phase.

- the description is correct and the ES agrees
- the description is wrong and the ES catches the error
- the description is wrong but the ES does not catch it
- the description is correct but the ES reports an error

The first two are acceptable; the last two are not. Let alpha represent the probability that the ES does not catch a bad component, and let beta represent the probability that the ES incorrectly tags a valid component. Alpha and beta can be reduced by a careful selection of the rules involved in the validation phase. In practice, however, they will not be zero: bad components will slip through the system, and the ES will object to some correct components.

To help alleviate the latter situation, the ES was designed so that the user could override this objection. The user is warned if the component does not meet the ES expectations, and he can either change the description if it is in error, or he can ignore the warning and continue. (Whenever the user overrides the ES, the contents of the current session is dumped to a file,

which can be viewed later by the expert. These "snapshots" of actual sessions can aid the expert in finding missing or incorrect rules in the knowledge base. More on this later.)

2. Derive the input parameters to COMPKIL

The main task of the ES is to feed COMPKIL the correct set of input parameters for the current component. Several of these parameters have natural default settings, see Appendix A, and they retain their default values unless the user requests to change them. The ES next displays the default input parameters to COMPKIL and asks the user if these are acceptable. The user can make any permissible changes to the default settings. There remain six groups of input parameters to be computed (for each of the six component faces):

- a. presented area,
- b. sensitive area,
- c. hole diameter required to render component nonfunctional,
- d. number of barriers inside the component,
- e. material type of each barrier and
- f. thickness of each barrier.

The ES uses a combination of engineering calculations and its knowledge base to derive the needed inputs. The following series of paragraphs describe, in more detail, the logic involved for any arbitrary component face, and the discussion is mainly concerned with the group of components called control rods (rods, shafts, cables, hydraulic cylinders, ...) whose main function is to transmit mechanical force from one location to another.

a. **Presented area.** For a given view, the presented area is defined to be the area of the projection of the component onto the face of an enclosing cube. See Figure 1. For the class of control rods, the projections are either rectangles or circles, and the presented areas can easily be computed from the component's physical description.

b. **Sensitive area.** The entire surface of a component face may not be susceptible to damage. There may be regions where a penetrator can inflict damage, yet the component will continue to function. The area of the component face which is potentially susceptible to damage is called the sensitive area. As an example, consider the tire shown in Figure 2. For a penetrator approaching from the side, the sensitive area is that portion

between the rim and the inside of the tread. A shot through the tread (from this direction) would not produce a flat tire. In general, the expert's judgement will determine which portion of the surface is sensitive.

c. Hole diameter. Any component under study performs some function essential to the continued operation of the vehicle. If that component is damaged to the extent that it can not perform its function, then it is said to be killed. This happens when a penetrator strikes a sensitive region and makes a hole large enough (and deep enough) to render it nonfunctional. The hole diameter is defined as the minimum hole size required in the sensitive area to kill the given component.

The function of components called control rods is to transmit mechanical force. If the amount of force to be transmitted by the component were known, then the hole size could be computed analytically given the material composition of the rod. Several factors currently militate against, or at least modify, this approach. First, the force imposed upon a rod will vary greatly under a given scenario. The force exerted by a vehicle's drive shaft will differ from smooth pavement to rough terrain. Second, damage to a component may have side effects. A small scratch or nick in the exposed part of a hydraulic cylinder may damage the seals and cause the fluid to leak out, thus rendering the cylinder nonfunctional. The expert's experience and engineering judgement are used to determine the hole diameter needed to kill the component.

d. Barriers, materials, thicknesses. These values are commonly taken from the component's description with no processing involved. There are cases for which this cannot be done. First, consider a hollow rod. The rod's inside dimension may be missing from the computer description, and so the expert must use his best judgement to estimate the wall thickness of that rod. Second, since COMPKIL only records complete penetrations, the expert must convert a "scratch" in a hydraulic cylinder to a penetration of an equivalent thickness of hard steel. Similarly, when the component can be defeated by a dent rather than a penetration, the expert must convert a "dent" into a penetration of an equivalent thickness of the component's material.

3. Run COMPKIL and prepare inputs for VAST

Once the six groups of input parameters described above have been determined, they are then displayed and the user has three options: 1) accept them; 2) ask for an explanation of how the parameters were computed; or 3) change them (thus bypassing the ES knowledge base). When the latter occurs, the current working memory is dumped to a file for later viewing by the expert.

After the user either has accepted the computed input parameters or has manually changed them, they are fed into COMPKIL. The outputs of COMPKIL are then displayed in tabular and in graphical form. If the results are not acceptable, the user can make various changes by returning to stage 1 or to stage 2 and repeating the cycle.

VI. SYSTEM DETAILS

At each stage the user has the option to cycle to the beginning of the current stage or to a previous one, in order to correct an input or to change a parameter. Whenever a change is made, the ES must "back up" to a previous state, update the appropriate elements of working memory, and run through some of its rules based on the new information in working memory. The "back-up" operations are usually transparent to the user.

An extensive help feature contains three levels of help information, and it can be invoked in several ways. The help feature is enabled automatically whenever the user gives an incorrect response to the ES. In addition, the user can type '?' to get specific help at any point in the session, or he can type '??' to get more general help about the ES. See Appendix C. There is also an explanation facility, so that the user can query the ES on how it performed its analysis. A sample of the user-ES dialog is given in Appendix A showing some of these features.

As mentioned earlier, a basic concept in the development of the ES is that it should only guide the user. The ES offers advice which can be accepted or ignored. Thus the rules used by the ES can be overridden if the user desires. Whenever this happens, the contents of working memory are dumped to a file. The expert periodically views this file to see if any of the rules need to be modified.

VII. CODE DETAILS

The ES uses routines written in several computer languages, including FORTRAN, C, and LISP. COMPKIL was written in FORTRAN; several C routines were written to handle friendly interfacing between the ES and the user; and much of the code was written in OPS5, which is a production system written at Carnegie Mellon University.

1. Working Memory

The data base in OPS5 (called working memory) consists of sequences of the form:

(x1 x2 x3 ...)

where the xi can be words or numbers. For example, during a sample run, the working memory consisted of the following:

```
(type      hollow      drive-shaft)
(component-name  drive shaft)
(component-number 123001)
(diam      2.5)
(length    60.0)
(vehicle   armored)
(material  1)
(power     mechanical)
(loading   torsion)
```

Working memory is used to store general facts about the domain and specific information about the current problem. It can also contain a list of tasks or goals to be accomplished. Elements in working memory can be created, destroyed, or modified.

2. Knowledge Base

The production rules in OPS5 have the form:

IF p-1 and p-2 and p-3 ... THEN q-1 and q-2 and ...

where each p-i is a pattern to be matched against the working memory, and each q-i is an action to modify the working memory by adding, changing, or deleting elements. Some sample rules from the ES will be presented in the next section.

The production rules form the "knowledge base" of the system. Each rule "codifies" a single node in the thought process used by the expert to solve the problem. In simplest terms, a rule says "if I see this, then I do this." If the expert's thought processes can be modeled by production rules, and if the entire set of rules can be captured, then the ES will perform just as the expert, over the given domain.

3. Sample Rules

Below are two rules used to predict the hole size for certain components. The first rule concerns components which are not hydraulic cylinders.


```

IF      (goal compute-hole-size)
        (power mechanical)
        (diam <d>)
not (type solid hyd-cylinder)
THEN
        (remove 1)
        (make hole-size-factor .50)
        (make hole-diam ( .50 * <d> ) )
        (make goal compute-compkil)

```

Thus if the goal is to compute hole size, and the power source is mechanical, with diameter d, and the object is not a hydraulic cylinder, then remove the first pattern (in this case "goal compute-hole-size") from working memory and make three new elements in working memory:

```

hole-size-factor .50
hole-diam .50*<d>
goal compute-compkil

```

Thus the minimum hole size needed to defeat this component is given by half the component's diameter. Note that the hole size factor used in this rule came from experimental data.

Now consider the following rule for hydraulic cylinders.

```

IF      (goal compute-hole-size)
        (power mechanical)
        (type solid hyd-cylinder)
THEN
        (remove 1)
        (make hole-size-factor scratch)
        (make hole-diam 0.125)
        (make goal compute-compkil)

```

If the goal is to compute hole size, and the power source is mechanical, and the object is a hydraulic cylinder, then remove the first pattern ("goal compute-hole-size") from working memory and make three new elements in working memory:

```

hole-size-factor scratch
hole-diam 0.125
goal compute-compkil

```

For this component, the kill criteria is a "scratch" to the cylinder; this will damage the seals, causing the fluid to leak out. In the expert's judgement, the minimum hole size to damage the seals is about 1/8 inch.

For more examples of production rules, see Appendix B.

VIII. CORPORATE MEMORY

The examples provided help to illustrate how knowledge can be coded into production rules. Several points need further discussion. First, the calculation of $P(k/h)$'s involves: 1) limited experimental data, 2) engineering calculations, and 3) a great deal of engineering judgement for those situations where data for formulas do not exist. Second, the production rule formalism provided a natural framework in which the expert could express his thoughts. Third, the body of rules elicited from the expert form a "corporate memory" that previously existed only in the mind of the expert. The lines of reasoning used by the expert are now permanently preserved, and can be viewed by others. Fourth, whenever new data or information becomes available, then the rules in the ES can be easily updated to incorporate this new knowledge.

IX. CURRENT STATUS

The ES for rods, shafts and cables has about 100 OPS5 rules. About 2/3 of these rules are used to detect errors in the component's description, and the remaining rules are used to compute the input parameters to COMPKIL. The rules were derived through hours of extensive interviews with the expert. The ES has gone through many revisions to modify incorrect rules and to add new rules as errors were uncovered. The current version runs on a VAX computer, and it is currently being used in a study of an Army combat vehicle. A sample of the knowledge base is included in Appendix B.

X. FUTURE DIRECTIONS

Work is underway to expand the knowledge base to include additional types of components. A prototype ES for fuel systems is now undergoing testing, and will soon expand to cooling systems, lubrication systems, hydraulic systems, electrical systems and communication systems. In addition to expanding the knowledge base, work is in progress to improve the interaction with the user. This work includes incorporating a windows package to display multiple screens, and a terminal independent three dimensional graphics capability for viewing the output of each COMPKIL run.

At present the ES must be told whether a component is critical to the sustained operation of the vehicle. Clearly an uninterrupted supply of fuel is needed for sustained operation, but not all fuel components are critical. The gas cap clearly is not; a reserve fuel cell may or may not be critical. It may be possible to extend the ES so that it can decide which components are critical, and this, too, is being investigated.

XI. CONCLUSION

The intent of this ES project is not to produce a computer program to replace humans in the stream of component defeat analysis, rather the intent is to write an ES that can accurately reproduce much of the routine work (both mathematical and logical) required for that analysis. Such an ES can reduce the time needed to produce P K/H probabilities and can also be used to train new individuals to perform component analyses and, most importantly, lend a higher degree of consistency to sets of analyses.

APPENDIX A

Sample ES Session

Notes:

- 1) User responses are given after the prompt >>>-->
- 2) Comments are given in parenthesis after some responses.

Welcome to BRL P K/H Expert System
for rods, shafts and cables

(Test Version 3.1)

Mon Apr 1 09:33:12 1985

>>>>> for help at any time type '?' <<<<<

*** do you want instructions ?

>>>--> n (<- See Appendix C for instructions.)

```
-
-
-
-
-
-
type in one of the choices
-
-
-
-
-
-
'sc' : solid control linkage
'sd' : solid drive shaft
'ss' : solid steering linkage
'sh' : solid hydraulic cylinder rod
'st' : solid torsion bar
'hc' : hollow control linkage
'hd' : hollow drive shaft
'cc' : cable control linkaged
'cb' : cable brake linkage
```

>>>--> hd

>>enter component name : >>>--> drive shaft

>>enter component number : >>>--> 123001

>>enter diameter in inches : >>>--> 2.5

>>enter length in inches : >>>--> 48

>>enter vehicle type, armored or light (a or l) : >>>--> a

>>enter material code (1-20) :

>>>--> ? (<- User typed ? for help.)

The material code numbers are:

1 = mild homogeneous steel	11 = unbounded nylon
2 = hard homogeneous steel	12 = bonded nylon
3 = face-hardened steel	13 = lexan
4 = cast iron	14 = plexiglas as cast
5 = aluminum alloy 2024t-3	15 = stretched plexiglas
6 = magnesium	16 = doron
7 = copper	17 = bullet-resistant glass
8 = lead	18 = rubber
9 = titanium alloy	19 = hard wood
10 = tuballoy	20 = water or person

>>enter material code (1-20) :

>>>--> 8 (<- Note this choice is unrealistic.)

>>enter power source, human or mechanical (h or m) :

>>>--> g (<- Typing error.)

>>input error...type ? for help

>>enter power source, human or mechanical (h or m) :

>>>--> m

- - - - -
>>enter loading code :
(1 2 3 4)

- - - - -
'1' : tension
'2' : compression
'3' : both tension and compression
'4' : torsion

>>>--> 4

***** warning ***** (<- Warning is given.)

>>> some of your inputs do not match my expectations <<<
>>> for a hollow control rod, I expect the following :
*** length/diam > = 30
...you typed length = 48.0 and diam = 2.5
giving a ratio = 19.2
*** material type = 1 or 2 or 3
... you typed 8

***** your inputs were *****
(1) object type = hollow drive-shaft
(2) length = 48.0
(3) diameter = 2.5
(4) material code = 8
(5) power source = mechanical
(6) loading = torsion
(7) vehicle type = armored

**** are these ok ?

>>>--> n (<- Chance to change inputs.)

Type the number(s) of the line(s) to change.

ONE number per line. Type 0 to exit

line to change: >>>--> 2

line to change: >>>--> 4

line to change: >>>--> 0

>>enter material code (1-20) : >>>--> 1

>>enter length in inches : >>>--> 60

***** warning *****
>>> some of your inputs do not match my expectations <<<
>>> for a hollow control rod, I expect the following :
*** length/diam > = 30
...you typed length = 60.0 and diam = 2.5
giving a ratio = 24.0

```

***** your inputs were *****
(1) object type = hollow drive-shaft
(2) length      = 60.0
(3) diameter    = 2.5
(4) material code = 1
(5) power source = mechanical
(6) loading      = torsion
(7) vehicle type = armored

```

**** are these ok ?

>>>--> y (<- User decides to override the warnings.)

*** current default settings for COMPKIL ***

```

1) fragment shape factor = 0.01103 improved
2) attack condition      = 1 random
3) max velocity          = 7000 feet per second
4) step                  = 4 4-step step functions
5) number of masses      = 12
   masses (in grains)
       1      3      5      10      15      30
       60     120    240    500    1000    5000
6) ktype                 = "mobility "
7) metric flag           = 0 data in english units
8) print flag            = 1 generate printed output
9) VAST output flag      = 1 generate output for VAST
10) title                = BRL pkh expert system
11) component subsection = 1

```

** are these ok ? >>>--> y

***** parameters for compkil *****

>>>> for all attack angles <<<<

1. hole diameter = 1.25
2. Ap = 6.25
3. As = 1.5625
4. material code = 1
5. thickness = 0.25

- - - - -
*** are these ok ?
(y or n or e)

- - - - -
'y' : yes
'n' : no
'e' : explain

>>>--> ? (<- User types ? for help.)

type y if these are ok, type n if you want to change them,
type e if you want an explanation

- - - - -
*** are these ok ?
(y or n or e)

- - - - -
'y' : yes
'n' : no
'e' : explain

>>>--> y

***** compkil results *****

input data for " BRL pkh expert system "

component number - 123001

component name - drive shaft

"mobility "

css	view	dhreq	hreq	nb	m(1)	th(1)	pa	sa
1	1	1.25	1.23	1	1	0.25	6.25	1.56
1	2	1.25	1.23	1	1	0.25	6.25	1.56
1	3	1.25	1.23	1	1	0.25	6.25	1.56
1	4	1.25	1.23	1	1	0.25	6.25	1.56
1	5	1.25	1.23	1	1	0.25	6.25	1.56
1	6	1.25	1.23	1	1	0.25	6.25	1.56

(u) conditional kill probabilities for
random, single fragment attack

fragment mass (grs)	(gms)	striking velocity (ft/sec)	(m/sec)	pk/h

5000.00	323.99	1000	305	0.05
		1400	427	0.21

cutoff mass = 1173.55 grains (76.045 grams)				

*** is the output ok ? >>>--> y

*** do you wish to continue ? >>>--> n

**** BRL P K/H Expert System signing off ****

APPENDIX B

Sample Rules

```

;
; ***** check user inputs *****
;
; ***** check ratio *****
(p check-ratio-cable-bad
  (check ratio)
  (type cable)
  (ratio < 48)
  -->
  (remove 1)
  (make warning ratio cable)
)
(p check-ratio-cable-ok
  (check ratio)
  (type cable)
  (ratio >= 48)
  -->
  (remove 1)
)
(p check-ratio-solid-bad-high
  (check ratio)
  (type solid)
  (ratio > 30 )
  -->
  (remove 1)
  (make warning ratio solid)
)
(p check-ratio-solid-bad-low
  (check ratio)
  (type solid)
  (ratio <= 4 )
  -->
  (remove 1)
  (make warning ratio solid)
)
(p check-ratio-solid-ok
  (check ratio)
  (type solid)
  (ratio { > 4 <= 30 } )
  -->
  (remove 1)
)
(p check-ratio-hollow-bad
  (check ratio)
  (type hollow)
  (ratio < 30)
  -->
  (remove 1)
  (make warning ratio hollow)
)
(p check-ratio-hollow-ok
  (check ratio)
  (type hollow)

```

```

    (ratio >= 30)
    -->
      (remove 1)
    )
; ***** check material *****
(p check-material-solid-bad
  (check material)
  (type solid)
  -(material << 1 2 3 >> )
  -->
    (remove 1)
    (make warning material solid)
  )
(p check-material-solid-ok
  (check material)
  (type solid)
  (material << 1 2 3 >> )
  -->
    (remove 1)
  )
(p check-material-hollow-bad
  (check material)
  (type hollow)
  -(material << 1 2 3 >> )
  -->
    (remove 1)
    (make warning material hollow)
  )
(p check-material-hollow-ok
  (check material)
  (type hollow)
  (material << 1 2 3 >> )
  -->
    (remove 1)
  )
(p check-material-cable-bad
  (check material)
  (type cable)
  -(material << 1 2 3 >> )
  -->
    (remove 1)
    (make warning material cable)
  )
(p check-material-cable-ok
  (check material)
  (type cable)
  (material << 1 2 3 >> )
  -->
    (remove 1)
  )
; ***** check diameter *****
(p check-diam-cable-human-bad
  (check diam)

```

```

(type cable)
(power human)
(diam > .125 )
-->
  (remove 1)
  (make warning diam cable)
)
(p check-diam-cable-human-ok
  (check diam)
  (type cable)
  (power human)
  (diam <= .125 )
  -->
    (remove 1)
  )
(p check-diam-cable-mechanical-bad
  (check diam)
  (type cable)
  (power mechanical)
  (diam < .375 )
  -->
    (remove 1)
    (make warning diam cable)
  )
(p check-diam-cable-mechanical-ok
  (check diam)
  (type cable)
  (power mechanical)
  (diam >= .375 )
  -->
    (remove 1)
  )
(p check-diam-solid-mechanical-bad
  (check diam)
  (type solid)
  (power mechanical)
  - (diam { > 1 < 3 } )
  -->
    (remove 1)
    (make warning diam solid)
  )
(p check-diam-solid-mechanical-ok
  (check diam)
  (type solid)
  (power mechanical)
  (diam { > 1 < 3 } )
  -->
    (remove 1)
  )
(p check-diam-solid-human-torsion-bad
  (check diam)
  (type solid)
  (power human)

```

```

    (loading torsion)
  - (diam { > .625 < .75 } )
  -->
    (remove 1)
    (make warning diam solid)
  )
(p check-diam-solid-human-torsion-ok
  (check diam)
  (type solid)
  (power human)
  (loading torsion)
  (diam { > .625 < .75 } )
  -->
    (remove 1)
  )
(p check-diam-solid-human-nontorsion-bad
  (check diam)
  (type solid)
  (power human)
  - (loading torsion)
  - (diam { > .125 < .375 } )
  -->
    (remove 1)
    (make warning diam solid)
  )
(p check-diam-solid-human-nontorsion-ok
  (check diam)
  (type solid)
  (power human)
  - (loading torsion)
  (diam { > .125 < .375 } )
  -->
    (remove 1)
  )
(p check-diam-hollow-mechanical-bad
  (check diam)
  (type hollow)
  (power mechanical)
  (diam < 2 )
  -->
    (remove 1)
    (make warning diam hollow)
  )
(p check-diam-hollow-mechanical-ok
  (check diam)
  (type hollow)
  (power mechanical)
  (diam >= 2 )
  -->
    (remove 1)
  )
(p check-diam-hollow-human-compression-bad
  (check diam)

```

```

(type hollow)
(power human)
(loading compression)
- (diam { > .25 < .5 } )
-->
  (remove 1)
  (make warning diam hollow)
)
(p check-diam-hollow-human-compression-ok
  (check diam)
  (type hollow)
  (power human)
  (loading compression)
  (diam { > .25 < .5 } )
-->
  (remove 1)
)
(p check-diam-hollow-human-tension-bad
  (check diam)
  (type hollow)
  (power human)
  (loading tension)
  - (diam { > .25 < .5 } )
-->
    (remove 1)
    (make warning diam hollow)
  )
(p check-diam-hollow-human-tension-ok
  (check diam)
  (type hollow)
  (power human)
  (loading tension)
  (diam { > .25 < .5 } )
-->
    (remove 1)
  )
(p check-diam-hollow-human-both-bad
  (check diam)
  (type hollow)
  (power human)
  (loading both)
  - (diam { > .25 < .5 } )
-->
    (remove 1)
    (make warning diam hollow)
  )
(p check-diam-hollow-human-both-ok
  (check diam)
  (type hollow)
  (power human)
  (loading both)
  (diam { > .25 < .5 } )

```



```

-->
  (remove 1)
)
(p check-diam-hollow-human-torsion-bad
  (check diam)
  (type hollow)
  (power human)
  (loading torsion)
- (diam { > .75 < 1.0 } )
-->
  (remove 1)
  (make warning diam hollow)
)
(p check-diam-hollow-human-torsion-ok
  (check diam)
  (type hollow)
  (power human)
  (loading torsion)
  (diam { > .75 < 1.0 } )
-->
  (remove 1)
)

; ***** check loading *****
(p check-loading-solid-ok
  (check loading)
  (type solid)
-->
  (remove 1)
)
(p check-loading-cable-bad
  (check loading)
  (type cable)
- (loading tension)
-->
  (remove 1)
  (make warning loading cable )
)
(p check-loading-cable-ok
  (check loading)
  (type cable)
  (loading tension)
-->
  (remove 1)
)
(p check-loading-hollow-bad
  (check loading)
  (type hollow)
  (loading tension)
-->
  (remove 1)
  (make warning loading hollow )
)

```

```
(p check-loading-hollow-ok
  (check loading)
  (type hollow)
  -(loading tension)
  -->
  (remove 1)
  )
;
```

APPENDIX C

Help Feature

There are several types of commands that are available whenever you are being prompted for input. You can respond to the ">>--> " prompt in these ways:

Type in your response.

Type a '?' for help. Displays more information about the current question.

Type '?help' or '??' to print this message.

Type '!' followed by a shell command. for example,
 '!pwd' will print your current directory and
 '!ls' lists the files in your current directory.

Type '>' followed by a lisp command.
The two commands of interest are
[wm] and [snapshot "message"].
type '>[wm]' to see the contents of working memory.
type '>[snapshot "any message in double quotes"]'
to have working memory plus your message dumped to a file.

- - - - -
Note: If you get an error condition, you can exit by typing
CTRL-D.

also All responses require a carriage return.

DISTRIBUTION LIST

<u>No. of</u> <u>Copies</u>	<u>Organization</u>	<u>No. of</u> <u>Copies</u>	<u>Organization</u>
12	Administrator Defense Technical Info Center ATTN: DTIC-DDA Cameron Station Alexandria, VA 22304-6145	1	Director U.S. Army Air Mobility Research and Development Laboratory Ames Research Center Moffett Field, CA 94035
1	HQDA DAMA-ART-M Washington, DC 20310	1	Commander U.S. Army Communications - Electronics Command ATTN: AMSEL-ED Fort Monmouth, NJ 07703-5301
1	Commander U.S. Army Materiel Command ATTN: AMCDRA-ST 5001 Eisenhower Avenue Alexandria, VA 22333-0001	1	Commander ERADCOM Technical Library ATTN: DELSD-L (Reports Section) Fort Monmouth, NJ 07703-5301
1	Commander Armament R&D Center U.S. Army AMCCOM ATTN: SMCAR-TSS Dover, NJ 07801-5001	1	Commander US Army Missile Command Research, Development and Engineering Center ATTN: AMSMI-RD Redstone Arsenal, AL 35898
1	Commander Armament R&D Center U.S. Army AMCCOM ATTN: SMCAR-TDC Dover, NJ 07801-5001	1	Commander US Army Missile and Space Intelligence Center ATTN: AIAMS-YDL Redstone Arsenal, AL 35898-5500
1	Director Benet Weapons Laboratory Armament R&D Center U.S. Army AMCCOM ATTN: SMCAR-LCB-TL Watervliet, NY 12189	1	Commander U.S. Army Tank Automotive Command ATTN: AMSTA-TSL Warren, MI 48090
1	Commander U.S. Army Armament, Munitions and Chemical Command ATTN: SMCAR-ESP-L Rock Island, IL 61299	1	Director U.S. Army TRADOC Systems Analysis Activity ATTN: ATAA-SL White Sands Missile Range, NM 88002-5022
1	Commander U.S. Army Aviation Research and Development Command ATTN: AMSAV-E 4300 Goodfellow Blvd St. Louis, MO 63120	1	Commandant U.S. Army Infantry School ATTN: ATSH-CD-CSO-OR Fort Benning, GA 31905

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander U.S. Army Development and Employment Agency ATTN: MODE-TED-SAB Fort Lewis, WA 98433-5000
1	AFWL/SUL Kirtland AFB, NM 87117
1	Air Force Armament Laboratory ATTN: AFATL/DLODL Eglin AFB, NM 32542-5000

Aberdeen Proving Ground

Dir, USAMSAA
ATTN:AMXSY-D
AMXSY-MP, H. Cohen
Cdr, USATECOM
ATTN:AMSTE-TO-F
Cdr, CRDC, AMCCOM
ATTN:SMCCR-RSP-A
SMCCR-MU
SMCCR-SPS-IL

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. BRL Report Number _____ Date of Report _____
2. Date Report Received _____
3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.) _____

4. How specifically, is the report being used? (Information source, design data, procedure, source of ideas, etc.) _____

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided or efficiencies achieved, etc? If so, please elaborate. _____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.) _____

CURRENT
ADDRESS

Name

Organization

Address

City, State, Zip

7. If indicating a Change of Address or Address Correction, please provide the New or Correct Address in Block 6 above and the Old or Incorrect address below.

OLD
ADDRESS

Name

Organization

Address

City, State, Zip

(Remove this sheet along the perforation, fold as indicated, staple or tape closed, and mail.)